

Transient signal flagging algorithm definition for non-radiance data

Summary: This document provides the description of the transient signal flagging algorithm for *non-radiance* data to be implemented in the GDPS. The function is described with the associated input and output data. The OPF data is supplied for the required instrument configurations. The flagged transient pixels have to be excluded in the calculation of the irradiance averages.

		Date	Signature
Author:	Q. L. Kleipool	July 12, 2005	
Checked:	N. Rozemeijer	July 12, 2005	
Approved:	M. Dobber		
Approved:	B. van den Oord		
Archive:	R. Noordhoek		

DISTRIBUTION LIST

Pieterneel Levelt	KNMI
Marcel Dobber	KNMI
Bert van den Oord	KNMI
Nico Rozemeijer	TriOpSys
M. ter Linden	Dutch Space
V. Schenkelaars	Dutch Space
T. Watts	Dutch Space
Harry Förster	NIVR

CHANGE STATUS

Issue	Date	Author	Comments	Affected pages
1	June 8, 2005	Q. L. Kleipool	Initial version	All
2	July 12, 2005	Q. L. Kleipool	full OPF table added for all ICID and versions	12-48

References

- [rd_01] Wirth, Niklaus, *Algorithms + Data structures = Programs*, p. 84. Prentice Hall, 1976, isbn: 0130224189

Acronyms

CCD	:	Charge Coupled Device
NNM	:	Nearest Neighbourhood Method
OMI	:	Ozone Monitoring Instrument
OPF	:	Operational Parameter File
ROR	:	Read-Out Register
SAA	:	Southern Atlantic Anomaly
SDM	:	Sequential Difference Method
SNR	:	Signal to Noise Ratio
UV	:	Ultra Violet
VIS	:	Visual

1 Transient signal flagging algorithm for non-radiance data

The purpose of the algorithm is to remove spikes from OMI measurements of non-radiance data. These spikes are caused by radiation events, i.e., cosmic particles, protons and electrons encountered in space. Non-radiance data is here understood to be all measurements that are not earth-shine. Non-radiance data are WLS, LED and Sun irradiance measurements, as well as all dark measurements, i.e., all measurement classes SUN, WLS, LED and DARK.

1.1 Algorithm description

1.1.1 General description

The algorithm is designed to work on raw, unprocessed level 0 data, so no offset or gain corrections are required, i.e., executed after the RTS-possibility flagging and before co-addition division. The algorithm is based on the comparison of a image to the previous measurement, and referred to as the sequential difference method (SDM). Each measurement is divided by its predecessor to yield values close to unity for all pixels. Additional smoothing is then applied and spikes are easily detected by checking each pixel against certain thresholds. The smoothing is applied independently in the row and column direction. The aforementioned thresholds have been pre-calculated for each instrument configuration, and are stored in the operational parameter file (OPF).

For the first image a different approach has been developed because no predecessor image is available. For the first image a nearest neighbourhood method (NNM) is used to detect spikes. This is not as accurate as the sequential difference method, but yields acceptable results in the absence of other methods.

The aforementioned OPF parameters define the smooth width in the row direction, the smooth width in the column direction and a threshold in the row and column direction. These are defined separately for the NNM and SDM algorithm. Each of these sets are defined separately for the UV1, UV2 and VIS sub-channels. This yields a total of 21 parameters for each instrument configuration.

1.1.2 Applicable area

The NNM and SDM algorithm both work on all available data, i.e., the full image, but excluding the read-out-register (ROR) and the excess pixels. For the UV channel a distinction is made between the upper and lower wavelength part of the CCD; for the VIS channel no such distinction is made. The UV channel is separated at column 615 (zero based), with UV2 all columns below 615 and UV1 all columns above and including 615. The distinction is only made for certain measurement classes as specified in the following table:

Measurement class	Description	Number of sub-channels	Split column number
0	EARTH	2	615
1	SUN	2	615
2	WLS	2	615
3	LED	1	n/a
4	DARK	1	n/a

1.1.3 Flagged data handling

Pixels that have one of the following flags raised should be ignored by the NNM and SDM algorithms:

- **Dead Pixel**
- **Missing Pixel**
- **Processing Error**

1.1.4 Main driver routine

1.1.4.1 Computational sequence

- Determine the instrument configuration ID and the version number. Use this combination to extract the appropriate OPF data.
- Determine the measurement class of the current image, and use this to decide on the number of [sub]-channels required.
- For the VIS channel only a single sub-channel is defined. Call the transient algorithm with the complete image and the OPF data identified with 'VIS'.
- Depending on the measurement class, the UV channel can be separated in two sub-channels. If this is the case then call the transient routine for each sub-channel with its appropriate settings identified by 'UV1' and 'UV2'. If this is not the case, call the transient algorithm for the whole image with the settings for 'UV1'. This holds for both the NNM and SDM algorithms.
- Check if the current image is the first image of a sequence belonging to a single instrument configuration and version. Especially ensure that no problems arise from mixed-mode states during switches between instrument configurations. Also check that the number of rows and columns in both images are equal. If not store the signal data in a buffer and call the nearest neighbourhood method to determine the transient flags. If true, call the sequential difference algorithm to determine the transient flags.

1.1.5 Nearest neighbourhood method

1.1.5.1 Computational sequence

- Check whether the requested OPF column smooth width does not lead to median widths that are larger than the amount of columns in the image. The median width shall never be larger than the number of columns in the image.
- Check whether the requested OPF row smooth width does not lead to median widths that are larger than the amount of rows in the image. The median width shall never be larger than the number of rows in the image.
- Loop over all pixels in the input image, that are not flagged according section 1.1.3.
- For each pixel extract its neighbour pixel values; the OPF width indicates how many surrounding pixels should be included in the row and column direction independently. A value of zero indicates that only the centre pixel is included. A value of one indicates one higher and one lower (in total three pixels) to include, etc. This means that when the OPF widths in both direction equals one a total of nine pixels are used (3x3). At the edges of the image the full width must be used. If due to pixel flagging no pixels remain, no transient flag can be calculated for this pixel, and proceed with the next pixel.
- Calculate the median value of the pixels of the previous step.
- Divide the pixel under investigation by this median value and subtract one.
- If this value is greater than the requested threshold than raise the transient flag for that pixel.
- Repeat for all pixels

1.1.5.2 Equations

$$c = \text{opf_column_width}(w)$$

$$r = \text{opf_row_width}(w)$$

$$\text{spike_level}(i, j) = \frac{\text{signal}(i, j)}{\text{median}(\text{signal}(i - c \dots i + c, j - r \dots j + r))} - 1$$

$$\text{transient_flag}(i, j) = \text{spike_level}(i, j) \geq \text{opf_threshold}(w)$$

1.1.5.3 Definition of variables

The description of the variables are given in Table 2 to Table 3. The following abbreviations are used in the tables. Column C describes the characteristic of the variable (I=Input, L=Local, O=Output). Column D describes the type (df=float, ui=unsigned integer). Column U describes the dimension (dl=dimensionless).

variables	descriptive name	C	D	U	range	reference
signal(i,j)	input signal	I	df	adu		
w	instrument configuration and version identifier	I	ui	dl		
opf_threshold	threshold	I	df	dl		operational parameter file
opf_column_width		I	ui	dl		operational parameter file
opf_row_width		I	ui	dl		operational parameter file

Table 1: Input variables associated with the transient signal flagging using the nearest neighbourhood algorithm

variables	descriptive name	C	D	U	range	reference
i	column number	L	ui	dl		
j	row number	L	ui	dl		
r	OPF row width	L	ui	dl		
c	OPF column width	L	ui	dl		
spike_level(i,j)	to be compared to the thresholds	L	df	dl		

Table 2: Local variables associated with the transient signal flagging using the nearest neighbourhood algorithm

variables	descriptive name	C	D	U	range
transient_flag(i,j)	Transient flag	O	ui	dl	true, false

Table 3: Output variables associated with the transient signal flagging using the nearest neighbourhood algorithm

1.1.5.4 Properties of the NNM median filter

- If the number of input pixels is odd, the median value is the middle number when the array is sorted.
- If the number of input pixels is even, the median value is the mean of the two middle numbers when the array is sorted.
- At the edges, the filter should fill all values from the edge to the width with the median value of the width. Example: if the OPF width equals two, than the median width equals five and consequently all five values at the edge are filled with the median value of these five elements; note that all these five elements are same. Note that this deviates from regular implementations of median algorithms. This feature is required to ensure that spikes at the edges can be found.
- If the OPF median width in either direction is less than zero, then take the median of the whole array in that direction.
- If the OPF median width in either direction is greater than the image size in that direction filter over all pixels in that direction.
- The algorithm by Wirth to calculate the median is recommended [rd_01].

1.1.6 Sequential difference method

1.1.6.1 Computational sequence

- Pixel-wise divide the current image by the previous image which is stored in the buffer. Prevent division by zero by padding the resulting ratio (a.k.a. spike-level) with ones (1.0) if necessary. Also pad all pixels for which one of the flags specified in section 1.1.3 is raised.
- Check whether the OPF column smooth width is less than or equal to the number of columns in the input image. If not set the smooth width equal to the number of columns in the image.
- Check whether the OPF row smooth width is less than or equal to the number of rows in the input image. If not set the smooth width equal to the number of rows in the image.
- Loop over all rows in the spike-level image. Use a median filter to calculate a smooth version of each row of the spike-level data. The width of this filter varies per instrument configuration and is set in the OPF. If the OPF median width is set to zero, than skip this step altogether.
- Divide the current row by the smoothed version of the row. Check for division by zero; if necessary pad with ones (1.0)
- Loop over all pixels in the current row. For each pixel in this row determine whether the value calculated minus one exceeds the allowed spike threshold as set by the OPF, and store this row-flag locally.

- Consecutively loop over all columns in the image. Use a median filter to calculate a smooth version of each column of spike-level data. The width of this filter varies per instrument configuration and is set in the OPF. If the OPF median width is set to zero, than skip this step altogether.
- Divide the current column by the smoothed version of the column. Check for division by zero; if necessary pad with ones (1.0)
- Loop over all pixels in the current column. For each pixel in column row determine whether the value calculated minus one exceeds the allowed spike threshold as set by the OPF, and store this column flag locally.
- Loop over all pixels in the image. Flag all pixels as transient when both the row-flag and the column-flag has been raised (logical AND function).

1.1.6.2 Equations

$$\text{normalized_signal}(i, j, m) = \frac{\text{signal}(i, j, m, f)}{\text{signal}(i, j, m, f - 1)}$$

$$\text{median_filtered_col}(i, j, m) = \text{MEDIAN}(\text{normalized_signal}(all, j, m), \text{opf_median_width_col}(m))$$

$$\text{median_filtered_row}(i, j, m) = \text{MEDIAN}(\text{normalized_signal}(i, all, m), \text{opf_median_width_row}(m))$$

$$\text{spike_level_col}(i, j, m) = \frac{\text{normalized_signal}(i, j, m)}{\text{median_filtered_col}(i, j, m)} - 1$$

$$\text{spike_level_row}(i, j, m) = \frac{\text{normalized_signal}(i, j, m)}{\text{median_filtered_row}(i, j, m)} - 1$$

$$\text{spike_flag_row}(i, j, m) = \text{spike_level_row}(i, j, m) \geq \text{opf_spike_threshold_row}(m)$$

$$\text{spike_flag_col}(i, j, m) = \text{spike_level_col}(i, j, m) \geq \text{opf_spike_threshold_col}(m)$$

$$\text{transient_flag}(i, j, m) = \text{spike_flag_row}(i, j, m) \wedge \text{spike_flag_col}(i, j, m)$$

1.1.6.3 Definition of variables

variables	descriptive name	C	D	U	range	reference
signal(i,j,m,f)	earths radiance measurement	I	df	adu		
opf_median_width_col(m)	width of the median filter in the column dimension	I	ui	dl		Operational parameter file
opf_median_width_row(m)	width of the median filter in the row dimension	I	ui	dl		Operational parameter file
opf_spike_threshold_row(m)	spike threshold	I	df	dl		Operational parameter file
opf_spike_threshold_col(m)	spike threshold	I	df	dl		Operational parameter file

Table 4: Input variables associated with the SDM algorithm

Variables	descriptive name	C	D	U	range
i	column number in the optic region	L	ui	dl	
j	row number in the optic region	L	ui	dl	
w	instrument configuration and version identifier	L	ui	dl	
m	sub-channel identifier	L	ui	dl	UV1, UV2, VIS
f	sequential image identifier	L	ui	dl	
normalized_signal(i,j,m)	normalized signal	L	df	dl	
median_filtered_row(i,j,m)	median filtered row data	L	df	dl	
median_filtered_col(i,j,m)	median filtered column data	L	df	dl	
spike_level_col(i,j,m)	will be compared to the column thresholds	L	df	dl	
spike_level_row(i,j,m)	will be compared to the row thresholds	L	df	dl	
spike_flag_col(i,j,m)		L	ui	dl	
spike_flag_row(i,j,m)					

Table 5: Local variables associated with the SDM algorithm

variables	descriptive name	C	D	U	range
transient_flag(i,j)	transient flag	O	ui	dl	true, false

Table 6: Output variables associated with the SDM algorithm

1.1.6.4 Properties of the SDM boxcar median filter

- The median filter works like a boxcar average, with the difference that the filtered value are not calculated by the mean, but by the median.
- The width of the median filter is the amount of pixels that are included in the calculation of the median value.
- If the width is an odd number, the median value is the middle number when the array is sorted.
- If the width is an even number, the median value is the mean of the two middle numbers when the array is sorted.
- At the edges, the filter should fill all values from the edge to the width with the median value of the width. Example: if the width equals five than all five values at the edge are filled with the median value of these five elements; note that all these five elements are same. Note that this deviates from regular implementations of median algorithms. This feature is required to ensure that spikes at the edges can be found.
- If the OPF median width equals zero or one than no filtering will be applied to the input array
- If the OPF median width is less than zero take the median of the whole array
- If the OPF median width is greater than one then smooth the row data using the specified width
- The algorithm for the calculation of the median by Wirth is recommended [RD_01].

1.2 Comments on prototype code

The IDL prototype code is attached in Appendix A. This code ignores the pipeline environment of the actual data processor. It is only supplied to clarify the algorithm description and to be used as an example. IDL codes are overloaded to support vector data, so no explicit references to row or column numbers are needed. A star (*) used in the indexing of an array denotes that the operation works on all elements of the array in that specific dimension.

2 OPF data for transient signal flagging

The OPF data will be supplied for each (sub)channel separately and consist of two tables per channel. The first table contains the instrument configuration ID and version numbers of all states together with the median filter widths. The second table only contains the threshold for each filter. The first table only contains integer data, the second only floating point data.

2.1 OPF data for the UV1 channel

IcId	version	NNM_c	NNM_r	SDM_c	SDM_r
3	0	1	1	11	11
3	1	1	1	11	11
3	2	1	1	11	11
4	0	1	1	11	11
4	1	1	1	11	11
4	2	1	1	11	11
5	0	1	1	11	11
5	1	1	1	11	11
5	2	1	1	11	11
6	0	1	1	11	11
6	1	1	1	11	11
6	2	1	1	11	11
8	0	1	1	11	11
8	1	1	1	11	11
8	2	1	1	11	11
8	3	1	1	11	11
9	0	1	1	11	11
9	1	1	1	11	11
9	2	1	1	11	11
9	3	1	1	11	11
10	0	1	1	11	11
11	0	1	1	11	11
12	0	1	1	11	11
13	0	1	1	11	11

IcId	version	NNM_c	NNM_r	SDM_c	SDM_r
14	0	1	1	11	11
14	1	1	1	11	11
14	2	1	1	11	11
15	0	1	1	11	11
15	1	1	1	11	11
15	2	1	1	11	11
16	0	1	1	11	11
17	0	1	1	11	11
18	0	1	1	11	11
18	1	1	1	11	11
18	2	1	1	11	11
18	3	1	1	11	11
19	0	1	1	11	11
19	1	1	1	11	11
19	2	1	1	11	11
19	3	1	1	11	11
20	0	1	1	11	11
21	0	1	1	11	11
22	0	1	1	11	11
22	1	1	1	11	11
22	2	1	1	11	11
23	0	1	1	11	11
23	1	1	1	11	11
23	2	1	1	11	11
24	0	1	1	11	11
24	1	1	1	11	11
24	2	1	1	11	11
25	0	1	1	11	11
25	1	1	1	11	11
25	2	1	1	11	11

IcId	version	NNM_c	NNM_r	SDM_c	SDM_r
26	0	1	1	11	11
27	0	1	1	11	11
28	0	1	1	11	11
29	0	2	0	11	11
30	0	1	1	11	11
30	1	1	1	11	11
30	2	1	1	11	11
30	3	1	1	11	11
31	0	1	1	11	11
31	1	1	1	11	11
31	2	1	1	11	11
31	3	1	1	11	11
32	0	1	1	11	11
33	0	1	1	11	11
34	0	1	1	11	11
35	0	1	0	11	11
36	0	1	1	11	11
37	0	1	1	11	11
38	0	3	0	11	11
39	0	1	1	11	11
45	0	1	1	11	11
45	1	1	1	11	11
45	2	1	1	11	11
46	0	1	1	11	11
46	1	1	1	11	11
46	2	1	1	11	11
47	0	1	1	11	11
47	1	1	1	11	11
47	2	1	1	11	11
47	3	1	1	11	11

IcId	version	NNM_c	NNM_r	SDM_c	SDM_r
48	0	1	1	11	11
48	1	1	1	11	11
50	0	1	1	11	11
50	1	1	1	11	11
50	2	1	1	11	11
50	3	1	1	11	11
51	0	1	1	11	11
51	1	1	1	11	11
51	2	1	1	11	11
51	3	1	1	11	11
52	0	1	1	11	11
52	1	1	1	11	11
52	2	1	1	11	11
53	0	1	1	11	11
53	1	1	1	11	11
53	2	1	1	11	11
54	0	1	1	11	11
54	1	1	1	11	11
54	2	1	1	11	11
54	3	1	1	11	11
55	0	1	1	11	11
55	1	1	1	11	11
59	0	1	1	11	11
59	1	1	1	11	11
60	0	1	1	11	11
60	1	1	1	11	11
61	0	1	1	11	11
61	1	1	1	11	11
62	0	1	1	11	11
62	1	1	1	11	11

IcId	version	NNM_c	NNM_r	SDM_c	SDM_r
62	2	1	1	11	11
62	3	1	1	11	11
62	4	1	1	11	11
63	0	1	1	11	11
63	1	1	1	11	11
63	2	1	1	11	11
63	3	1	1	11	11
63	4	1	1	11	11
64	0	1	1	11	11
64	1	1	1	11	11
65	0	1	1	11	11
65	1	1	1	11	11
66	0	1	1	11	11
66	1	1	1	11	11
68	0	1	1	11	11
68	1	1	1	11	11
68	2	1	1	11	11
68	3	1	1	11	11
68	4	1	1	11	11
69	0	1	1	11	11
69	1	1	1	11	11
69	2	1	1	11	11
69	3	1	1	11	11
69	4	1	1	11	11
70	0	1	1	11	11
70	1	1	1	11	11
70	2	1	1	11	11
70	3	1	1	11	11
71	0	1	1	11	11
71	1	1	1	11	11

IcId	version	NNM_c	NNM_r	SDM_c	SDM_r
71	2	1	1	11	11
71	3	1	1	11	11
72	0	1	1	11	11
72	1	1	1	11	11
73	0	1	1	11	11
73	1	1	1	11	11
74	0	1	1	11	11
74	1	1	1	11	11
75	0	1	1	11	11
75	1	1	1	11	11
96	0	1	1	11	11
96	1	1	1	11	11
97	0	1	1	11	11
97	1	1	1	11	11
98	0	1	1	11	11
98	1	1	1	11	11
99	0	1	1	11	11
99	1	1	1	11	11
100	0	1	1	11	11
100	1	1	1	11	11
101	0	1	1	11	11
101	1	1	1	11	11
102	0	1	1	11	11
102	1	1	1	11	11
103	0	1	1	11	11
103	1	1	1	11	11
104	0	1	1	11	11
105	0	1	1	11	11
106	0	1	1	11	11
107	0	1	1	11	11

IcId	version	NNM_c	NNM_r	SDM_c	SDM_r
108	0	1	1	11	11
109	0	1	1	11	11
110	0	1	1	11	11
111	0	1	1	11	11
112	0	1	1	11	11
113	0	1	1	11	11
114	0	1	1	11	11
115	0	1	1	11	11
117	0	1	1	11	11
117	1	1	1	11	11
117	2	1	1	11	11
119	0	1	1	11	11
119	1	1	1	11	11
119	2	1	1	11	11
121	0	1	1	11	11
121	1	1	1	11	11
121	2	1	1	11	11
123	0	1	1	11	11
123	1	1	1	11	11
127	0	1	1	11	11
127	1	1	1	11	11
127	2	1	1	11	11
131	0	1	1	11	11
131	1	1	1	11	11
131	2	1	1	11	11
135	0	1	1	11	11
135	1	1	1	11	11
135	2	1	1	11	11
139	0	1	1	11	11
139	1	1	1	11	11

IcId	version	NNM_c	NNM_r	SDM_c	SDM_r
141	0	1	1	11	11
142	0	1	1	11	11

Table 7: OPF filter width parameters for the UV1 channel

NNM_cr	SDM_c	SDM_r
0.030	0.015	0.015
0.030	0.015	0.015
0.030	0.015	0.015
0.030	0.010	0.010
0.030	0.010	0.010
0.030	0.010	0.010
0.070	0.010	0.010
0.070	0.010	0.010
0.070	0.010	0.010
0.070	0.010	0.010
0.070	0.010	0.010
0.070	0.010	0.010
0.070	0.010	0.010
0.070	0.010	0.010
0.070	0.010	0.010
0.400	0.025	0.025
0.400	0.025	0.025
0.400	0.025	0.025
0.400	0.025	0.025
0.200	0.025	0.025
0.200	0.025	0.025
0.200	0.025	0.025
0.200	0.020	0.020
0.200	0.015	0.015
1.500	9.999	9.999
1.500	9.999	9.999
0.150	0.020	0.020
0.150	0.020	0.020
0.150	0.020	0.020
0.200	0.020	0.020
0.200	0.020	0.020
0.200	0.020	0.020
1.500	9.999	9.999
1.000	9.999	9.999
0.500	0.030	0.030
0.500	0.030	0.030
0.500	0.030	0.030
0.100	0.015	0.015

NNM_cr	SDM_c	SDM_r
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.150	0.015	0.015
0.050	0.015	0.015
0.030	0.015	0.015
0.030	0.015	0.015
0.030	0.015	0.015
0.030	0.010	0.010
0.030	0.010	0.010
0.070	0.010	0.010
0.070	0.010	0.010
0.070	0.010	0.010
0.070	0.010	0.010
0.070	0.010	0.010
0.200	0.030	0.030
0.900	0.020	0.020
0.040	0.015	0.015
0.040	0.015	0.015
0.300	0.015	0.015
0.300	0.015	0.015
0.300	0.015	0.015
0.100	0.020	0.020
0.100	0.020	0.020
0.100	0.020	0.020
0.100	0.020	0.020
0.050	0.010	0.010
0.050	0.015	0.015
0.020	0.010	0.010
0.050	0.010	0.010
0.050	0.006	0.006
0.020	0.007	0.007
0.050	0.005	0.005
0.030	0.006	0.006
0.030	0.015	0.015
0.030	0.015	0.015
0.040	0.010	0.010
0.040	0.010	0.010
0.070	0.010	0.010

NNM_cr	SDM_c	SDM_r
0.070	0.010	0.010
0.070	0.010	0.010
0.070	0.010	0.010
0.070	0.010	0.010
0.070	0.010	0.010
0.200	0.020	0.020
0.200	0.020	0.020
0.200	0.020	0.020
0.200	0.025	0.025
0.200	0.025	0.025
0.200	0.025	0.025
0.200	0.025	0.025
0.030	0.015	0.015
0.030	0.015	0.015
0.030	0.015	0.015
0.040	0.010	0.010
0.040	0.010	0.010
0.040	0.010	0.010
0.070	0.010	0.010
0.070	0.010	0.010
0.070	0.010	0.010
0.070	0.010	0.010
0.070	0.010	0.010
0.070	0.010	0.010
0.030	0.015	0.015
0.030	0.015	0.015
0.030	0.015	0.015
0.030	0.015	0.015
0.030	0.015	0.015
0.400	0.025	0.025
0.400	0.025	0.025
0.400	0.025	0.025
0.400	0.025	0.025
0.200	0.025	0.025
0.200	0.025	0.025
0.200	0.025	0.025
0.200	0.025	0.025
0.030	0.015	0.015
0.030	0.015	0.015
0.030	0.015	0.015

NNM_cr	SDM_c	SDM_r
0.030	0.015	0.015
0.030	0.015	0.015
0.030	0.015	0.015
0.200	0.015	0.015
0.200	0.015	0.015
0.200	0.015	0.015
0.200	0.015	0.015
0.200	0.015	0.015
0.040	0.010	0.010
0.040	0.010	0.010
0.040	0.010	0.010
0.040	0.010	0.010
0.200	0.015	0.015
0.200	0.015	0.015
0.200	0.015	0.015
0.200	0.015	0.015
0.040	0.010	0.010
0.040	0.010	0.010
0.040	0.010	0.010
0.040	0.010	0.010
0.200	0.015	0.015
0.200	0.015	0.015
0.040	0.020	0.020
0.040	0.020	0.020
0.200	0.015	0.015
0.200	0.015	0.015
0.040	0.015	0.015
0.040	0.015	0.015
0.003	0.005	0.005
0.003	0.005	0.005
0.005	0.005	0.005
0.005	0.005	0.005
0.010	0.006	0.006
0.010	0.006	0.006
0.030	0.007	0.007
0.030	0.007	0.007
0.040	0.007	0.007
0.040	0.007	0.007
0.100	0.010	0.010
0.100	0.010	0.010
0.200	0.015	0.015
0.200	0.015	0.015
2.500	0.100	0.100

NNM_cr	SDM_c	SDM_r
2.500	0.100	0.100
0.030	0.015	0.015
0.030	0.015	0.015
0.020	0.015	0.015
0.030	0.012	0.012
0.015	0.015	0.015
0.010	0.015	0.015
0.060	0.020	0.020
0.020	0.020	0.020
0.060	0.020	0.020
0.020	0.010	0.010
0.060	0.020	0.020
0.020	0.010	0.010
0.010	0.005	0.005
0.010	0.005	0.005
0.010	0.005	0.005
0.010	0.005	0.005
0.010	0.005	0.005
0.010	0.005	0.005
0.010	0.005	0.005
0.005	0.005	0.005
0.005	0.005	0.005
0.005	0.005	0.005
0.005	0.005	0.005
0.005	0.005	0.005
0.008	0.005	0.005
0.008	0.005	0.005
0.010	0.005	0.005
0.010	0.005	0.005
1.500	9.999	9.999
5.000	9.999	9.999

Table 8: OPF filter threshold parameters for the UV1 channel

2.2 OPF data for the UV2 channel

IcId	version	NNM_c	NNM_r	SDM_c	SDM_r
3	0	1	1	11	11
3	1	1	1	11	11
3	2	1	1	11	11
4	0	1	1	11	11
4	1	1	1	11	11
4	2	1	1	11	11
5	0	1	1	11	11
5	1	1	1	11	11
5	2	1	1	11	11
6	0	1	1	11	11
6	1	1	1	11	11
6	2	1	1	11	11
8	0	1	1	11	11
8	1	1	1	11	11
8	2	1	1	11	11
8	3	1	1	11	11
9	0	1	1	11	11
9	1	1	1	11	11
9	2	1	1	11	11
9	3	1	1	11	11
10	0	1	1	11	11
11	0	1	1	11	11
12	0	1	1	11	11
13	0	1	1	11	11
14	0	1	1	11	11
14	1	1	1	11	11
14	2	1	1	11	11
15	0	1	1	11	11
15	1	1	1	11	11

IcId	version	NNM_c	NNM_r	SDM_c	SDM_r
15	2	1	1	11	11
16	0	1	1	11	11
17	0	1	1	11	11
18	0	1	1	11	11
18	1	1	1	11	11
18	2	1	1	11	11
18	3	1	1	11	11
19	0	1	1	11	11
19	1	1	1	11	11
19	2	1	1	11	11
19	3	1	1	11	11
20	0	1	1	11	11
21	0	1	1	11	11
22	0	1	1	11	11
22	1	1	1	11	11
22	2	1	1	11	11
23	0	1	1	11	11
23	1	1	1	11	11
23	2	1	1	11	11
24	0	1	1	11	11
24	1	1	1	11	11
24	2	1	1	11	11
25	0	1	1	11	11
25	1	1	1	11	11
25	2	1	1	11	11
26	0	1	1	11	11
27	0	1	1	11	11
28	0	1	1	11	11
29	0	1	1	11	11
30	0	1	1	11	11

IcId	version	NNM_c	NNM_r	SDM_c	SDM_r
30	1	1	1	11	11
30	2	1	1	11	11
30	3	1	1	11	11
31	0	1	1	11	11
31	1	1	1	11	11
31	2	1	1	11	11
31	3	1	1	11	11
32	0	1	1	11	11
33	0	1	1	11	11
34	0	1	1	11	11
35	0	1	0	11	11
36	0	1	1	11	11
37	0	1	1	11	11
38	0	3	0	11	11
39	0	1	1	11	11
45	0	1	1	11	11
45	1	1	1	11	11
45	2	1	1	11	11
46	0	1	1	11	11
46	1	1	1	11	11
46	2	1	1	11	11
47	0	1	1	11	11
47	1	1	1	11	11
47	2	1	1	11	11
47	3	1	1	11	11
48	0	1	1	11	11
48	1	1	1	11	11
50	0	1	1	11	11
50	1	1	1	11	11
50	2	1	1	11	11

IcId	version	NNM_c	NNM_r	SDM_c	SDM_r
50	3	1	1	11	11
51	0	1	1	11	11
51	1	1	1	11	11
51	2	1	1	11	11
51	3	1	1	11	11
52	0	1	1	11	11
52	1	1	1	11	11
52	2	1	1	11	11
53	0	1	1	11	11
53	1	1	1	11	11
53	2	1	1	11	11
54	0	1	1	11	11
54	1	1	1	11	11
54	2	1	1	11	11
54	3	1	1	11	11
55	0	1	1	11	11
55	1	1	1	11	11
59	0	1	1	11	11
59	1	1	1	11	11
60	0	1	1	11	11
60	1	1	1	11	11
61	0	1	1	11	11
61	1	1	1	11	11
62	0	1	1	11	11
62	1	1	1	11	11
62	2	1	1	11	11
62	3	1	1	11	11
62	4	1	1	11	11
63	0	1	1	11	11
63	1	1	1	11	11

IcId	version	NNM_c	NNM_r	SDM_c	SDM_r
63	2	1	1	11	11
63	3	1	1	11	11
63	4	1	1	11	11
64	0	1	1	11	11
64	1	1	1	11	11
65	0	1	1	11	11
65	1	1	1	11	11
66	0	1	1	11	11
66	1	1	1	11	11
68	0	1	1	11	11
68	1	1	1	11	11
68	2	1	1	11	11
68	3	1	1	11	11
68	4	1	1	11	11
69	0	1	1	11	11
69	1	1	1	11	11
69	2	1	1	11	11
69	3	1	1	11	11
69	4	1	1	11	11
70	0	1	1	11	11
70	1	1	1	11	11
70	2	1	1	11	11
70	3	1	1	11	11
71	0	1	1	11	11
71	1	1	1	11	11
71	2	1	1	11	11
71	3	1	1	11	11
72	0	1	1	11	11
72	1	1	1	11	11
73	0	1	1	11	11

IcId	version	NNM_c	NNM_r	SDM_c	SDM_r
73	1	1	1	11	11
74	0	1	1	11	11
74	1	1	1	11	11
75	0	1	1	11	11
75	1	1	1	11	11
96	0	1	1	11	11
96	1	1	1	11	11
97	0	1	1	11	11
97	1	1	1	11	11
98	0	1	1	11	11
98	1	1	1	11	11
99	0	1	1	11	11
99	1	1	1	11	11
100	0	1	1	11	11
100	1	1	1	11	11
101	0	1	1	11	11
101	1	1	1	11	11
102	0	1	1	11	11
102	1	1	1	11	11
103	0	1	1	11	11
103	1	1	1	11	11
104	0	1	1	11	11
105	0	1	1	11	11
106	0	1	1	11	11
107	0	1	1	11	11
108	0	1	1	11	11
109	0	1	1	11	11
110	0	1	1	11	11
111	0	1	1	11	11
112	0	1	1	11	11

IcId	version	NNM_c	NNM_r	SDM_c	SDM_r
113	0	1	1	11	11
114	0	1	1	11	11
115	0	1	1	11	11
117	0	1	1	11	11
117	1	1	1	11	11
117	2	1	1	11	11
119	0	1	1	11	11
119	1	1	1	11	11
119	2	1	1	11	11
121	0	1	1	11	11
121	1	1	1	11	11
121	2	1	1	11	11
123	0	1	1	11	11
123	1	1	1	11	11
127	0	1	1	11	11
127	1	1	1	11	11
127	2	1	1	11	11
131	0	1	1	11	11
131	1	1	1	11	11
131	2	1	1	11	11
135	0	1	1	11	11
135	1	1	1	11	11
135	2	1	1	11	11
139	0	1	1	11	11
139	1	1	1	11	11
141	0	1	1	11	11
142	0	1	1	11	11

Table 9: OPF filter width parameters for the UV2 channel

NNM_cr	SDM_c	SDM_r
0.030	0.015	0.015
0.030	0.015	0.015
0.030	0.015	0.015
0.050	0.015	0.015
0.050	0.015	0.015
0.050	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.400	0.025	0.025
0.400	0.025	0.025
0.400	0.025	0.025
0.200	0.025	0.025
0.200	0.025	0.025
0.200	0.025	0.025
0.200	0.025	0.025
0.200	0.025	0.025
0.300	0.030	0.030
0.200	0.015	0.015
1.500	9.999	9.999
1.500	9.999	9.999
0.150	0.020	0.020
0.150	0.020	0.020
0.150	0.020	0.020
0.200	0.020	0.020
0.200	0.020	0.020
0.200	0.020	0.020
2.000	9.999	9.999
0.750	9.999	9.999
0.500	0.030	0.030
0.500	0.030	0.030
0.500	0.030	0.030
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.150	0.015	0.015
0.100	0.015	0.015
0.030	0.015	0.015

NNM_cr	SDM_c	SDM_r
0.030	0.015	0.015
0.030	0.015	0.015
0.050	0.015	0.015
0.050	0.015	0.015
0.050	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.200	0.030	0.030
0.900	0.020	0.020
0.040	0.015	0.015
0.040	0.015	0.015
0.200	0.015	0.015
0.200	0.015	0.015
0.200	0.015	0.015
0.100	0.020	0.020
0.100	0.020	0.020
0.100	0.020	0.020
0.100	0.020	0.020
0.070	0.010	0.010
0.050	0.015	0.015
0.020	0.010	0.010
0.050	0.010	0.010
0.050	0.006	0.006
0.020	0.007	0.007
0.050	0.005	0.005
0.030	0.006	0.006
0.030	0.015	0.015
0.030	0.015	0.015
0.040	0.015	0.015
0.040	0.015	0.015
0.040	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.200	0.020	0.020

NNM_cr	SDM_c	SDM_r
0.200	0.020	0.020
0.200	0.020	0.020
0.200	0.020	0.020
0.200	0.025	0.025
0.200	0.025	0.025
0.200	0.025	0.025
0.200	0.025	0.025
0.030	0.015	0.015
0.030	0.015	0.015
0.030	0.015	0.015
0.040	0.015	0.015
0.040	0.015	0.015
0.040	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.030	0.015	0.015
0.030	0.015	0.015
0.030	0.015	0.015
0.030	0.015	0.015
0.030	0.015	0.015
0.050	0.015	0.015
0.050	0.015	0.015
0.400	0.025	0.025
0.400	0.025	0.025
0.400	0.025	0.025
0.400	0.025	0.025
0.400	0.025	0.025
0.200	0.025	0.025
0.200	0.025	0.025
0.200	0.025	0.025
0.200	0.025	0.025
0.030	0.015	0.015
0.030	0.015	0.015
0.030	0.015	0.015
0.030	0.015	0.015
0.060	0.015	0.015
0.060	0.015	0.015
0.200	0.015	0.015
0.200	0.015	0.015
0.200	0.015	0.015

NNM_cr	SDM_c	SDM_r
0.200	0.015	0.015
0.200	0.015	0.015
0.040	0.010	0.010
0.040	0.010	0.010
0.040	0.010	0.010
0.040	0.010	0.010
0.040	0.010	0.010
0.050	0.015	0.015
0.050	0.015	0.015
0.050	0.015	0.015
0.050	0.015	0.015
0.040	0.005	0.005
0.040	0.005	0.005
0.040	0.005	0.005
0.200	0.015	0.015
0.200	0.015	0.015
0.050	0.040	0.040
0.050	0.040	0.040
0.050	0.015	0.015
0.050	0.015	0.015
0.060	0.010	0.010
0.060	0.010	0.010
0.010	0.005	0.005
0.010	0.005	0.005
0.020	0.005	0.005
0.020	0.005	0.005
0.030	0.007	0.007
0.030	0.007	0.007
0.070	0.008	0.008
0.070	0.008	0.008
0.080	0.008	0.008
0.080	0.008	0.008
0.150	0.012	0.012
0.150	0.012	0.012
0.250	0.020	0.020
0.250	0.020	0.020
2.500	0.150	0.150
2.500	0.150	0.150
0.040	0.015	0.015
0.030	0.015	0.015
0.020	0.015	0.015
0.030	0.012	0.012
0.020	0.015	0.015

NNM_cr	SDM_c	SDM_r
0.020	0.015	0.015
0.060	0.020	0.020
0.020	0.020	0.020
0.060	0.020	0.020
0.020	0.010	0.010
0.060	0.020	0.020
0.020	0.012	0.012
0.010	0.010	0.010
0.010	0.010	0.010
0.010	0.010	0.010
0.020	0.010	0.010
0.020	0.010	0.010
0.020	0.010	0.010
0.040	0.010	0.010
0.040	0.010	0.010
0.040	0.010	0.010
0.040	0.010	0.010
0.040	0.010	0.010
0.040	0.010	0.010
0.015	0.010	0.010
0.015	0.010	0.010
0.015	0.010	0.010
0.015	0.010	0.010
0.015	0.010	0.010
0.015	0.010	0.010
0.030	0.010	0.010
0.030	0.010	0.010
0.030	0.010	0.010
0.040	0.010	0.010
0.040	0.010	0.010
1.500	9.999	9.999
5.000	9.999	9.999

Table 10: OPF filter threshold parameters for the UV2 channel.

2.3 OPF data for the VIS channel

IcId	version	NNM_c	NNM_r	SDM_c	SDM_r
3	0	1	1	11	11
3	1	1	1	11	11
3	2	1	1	11	11
4	0	1	1	11	11
4	1	1	1	11	11
4	2	1	1	11	11
5	0	1	1	11	11
5	1	1	1	11	11
5	2	1	1	11	11
6	0	1	1	11	11
6	1	1	1	11	11
6	2	1	1	11	11
8	0	1	1	11	11
8	1	1	1	11	11
8	2	1	1	11	11
8	3	1	1	11	11
9	0	1	1	11	11
9	1	1	1	11	11
9	2	1	1	11	11
9	3	1	1	11	11
10	0	1	1	11	11
11	0	1	1	11	11
12	0	1	1	11	11
13	0	1	1	11	11
14	0	1	1	11	11
14	1	1	1	11	11
14	2	1	1	11	11
15	0	1	1	11	11

IcId	version	NNM_c	NNM_r	SDM_c	SDM_r
15	1	1	1	11	11
15	2	1	1	11	11
16	0	1	1	11	11
17	0	1	1	11	11
18	0	1	1	11	11
18	1	1	1	11	11
18	2	1	1	11	11
18	3	1	1	11	11
19	0	1	1	11	11
19	1	1	1	11	11
19	2	1	1	11	11
19	3	1	1	11	11
20	0	1	1	11	11
21	0	1	1	11	11
22	0	1	1	11	11
22	1	1	1	11	11
22	2	1	1	11	11
23	0	1	1	11	11
23	1	1	1	11	11
23	2	1	1	11	11
24	0	1	1	11	11
24	1	1	1	11	11
24	2	1	1	11	11
25	0	1	1	11	11
25	1	1	1	11	11
25	2	1	1	11	11
26	0	1	1	11	11
27	0	1	1	11	11
28	0	1	1	11	11
29	0	1	1	11	11

IcId	version	NNM_c	NNM_r	SDM_c	SDM_r
30	0	1	1	11	11
30	1	1	1	11	11
30	2	1	1	11	11
30	3	1	1	11	11
31	0	1	1	11	11
31	1	1	1	11	11
31	2	1	1	11	11
31	3	1	1	11	11
32	0	1	1	11	11
33	0	1	1	11	11
34	0	1	1	11	11
35	0	1	0	11	11
36	0	1	1	11	11
37	0	1	1	11	11
38	0	3	0	11	11
39	0	1	1	11	11
45	0	1	1	11	11
45	1	1	1	11	11
45	2	1	1	11	11
46	0	1	1	11	11
46	1	1	1	11	11
46	2	1	1	11	11
47	0	1	1	11	11
47	1	1	1	11	11
47	2	1	1	11	11
47	3	1	1	11	11
48	0	1	1	11	11
48	1	1	1	11	11
50	0	1	1	11	11
50	1	1	1	11	11

IcId	version	NNM_c	NNM_r	SDM_c	SDM_r
50	2	1	1	11	11
50	3	1	1	11	11
51	0	1	1	11	11
51	1	1	1	11	11
51	2	1	1	11	11
51	3	1	1	11	11
52	0	1	1	11	11
52	1	1	1	11	11
52	2	1	1	11	11
53	0	1	1	11	11
53	1	1	1	11	11
53	2	1	1	11	11
54	0	1	1	11	11
54	1	1	1	11	11
54	2	1	1	11	11
54	3	1	1	11	11
55	0	1	1	11	11
55	1	1	1	11	11
59	0	1	1	11	11
59	1	1	1	11	11
60	0	1	1	11	11
60	1	1	1	11	11
61	0	1	1	11	11
61	1	1	1	11	11
62	0	1	1	11	11
62	1	1	1	11	11
62	2	1	1	11	11
62	3	1	1	11	11
62	4	1	1	11	11
63	0	1	1	11	11

IcId	version	NNM_c	NNM_r	SDM_c	SDM_r
63	1	1	1	11	11
63	2	1	1	11	11
63	3	1	1	11	11
63	4	1	1	11	11
64	0	1	1	11	11
64	1	1	1	11	11
65	0	1	1	11	11
65	1	1	1	11	11
66	0	1	1	11	11
66	1	1	1	11	11
68	0	1	1	11	11
68	1	1	1	11	11
68	2	1	1	11	11
68	3	1	1	11	11
68	4	1	1	11	11
69	0	1	1	11	11
69	1	1	1	11	11
69	2	1	1	11	11
69	3	1	1	11	11
69	4	1	1	11	11
70	0	1	1	11	11
70	1	1	1	11	11
70	2	1	1	11	11
70	3	1	1	11	11
71	0	1	1	11	11
71	1	1	1	11	11
71	2	1	1	11	11
71	3	1	1	11	11
72	0	1	1	11	11
72	1	1	1	11	11

IcId	version	NNM_c	NNM_r	SDM_c	SDM_r
73	0	1	1	11	11
73	1	1	1	11	11
74	0	1	1	11	11
74	1	1	1	11	11
75	0	1	1	11	11
75	1	1	1	11	11
96	0	1	1	11	11
96	1	1	1	11	11
97	0	1	1	11	11
97	1	1	1	11	11
98	0	1	1	11	11
98	1	1	1	11	11
99	0	1	1	11	11
99	1	1	1	11	11
100	0	1	1	11	11
100	1	1	1	11	11
101	0	1	1	11	11
101	1	1	1	11	11
102	0	1	1	11	11
102	1	1	1	11	11
103	0	1	1	11	11
103	1	1	1	11	11
104	0	1	1	11	11
105	0	1	1	11	11
106	0	1	1	11	11
107	0	1	1	11	11
108	0	1	1	11	11
109	0	1	1	11	11
110	0	1	1	11	11
111	0	1	1	11	11

IcId	version	NNM_c	NNM_r	SDM_c	SDM_r
112	0	1	1	11	11
113	0	1	1	11	11
114	0	1	1	11	11
115	0	1	1	11	11
117	0	1	1	11	11
117	1	1	1	11	11
117	2	1	1	11	11
119	0	1	1	11	11
119	1	1	1	11	11
119	2	1	1	11	11
121	0	1	1	11	11
121	1	1	1	11	11
121	2	1	1	11	11
123	0	1	1	11	11
123	1	1	1	11	11
127	0	1	1	11	11
127	1	1	1	11	11
127	2	1	1	11	11
131	0	1	1	11	11
131	1	1	1	11	11
131	2	1	1	11	11
135	0	1	1	11	11
135	1	1	1	11	11
135	2	1	1	11	11
139	0	1	1	11	11
139	1	1	1	11	11
141	0	1	1	11	11
142	0	1	1	11	11

Table 11: OPF filter width parameters for the VIS channel

NNM_cr	SDM_c	SDM_r
0.050	0.015	0.015
0.050	0.015	0.015
0.050	0.015	0.015
0.050	0.015	0.015
0.050	0.015	0.015
0.050	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.400	0.025	0.025
0.400	0.025	0.025
0.400	0.025	0.025
0.300	0.025	0.025
0.300	0.025	0.025
0.300	0.025	0.025
0.300	0.025	0.025
0.700	0.030	0.030
0.400	0.020	0.020
2.000	9.999	9.999
2.000	9.999	9.999
0.150	0.030	0.030
0.150	0.030	0.030
0.150	0.030	0.030
0.400	0.030	0.030
0.400	0.030	0.030
0.400	0.030	0.030

NNM_cr	SDM_c	SDM_r
1.500	9.999	9.999
0.750	9.999	9.999
0.500	0.030	0.030
0.500	0.030	0.030
0.500	0.030	0.030
0.500	0.030	0.030
0.200	0.025	0.025
0.200	0.025	0.025
0.200	0.025	0.025
0.050	0.015	0.015
0.050	0.015	0.015
0.050	0.015	0.015
0.050	0.015	0.015
0.050	0.015	0.015
0.050	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.350	0.050	0.050
0.500	0.020	0.020
0.040	0.015	0.015
0.040	0.015	0.015
0.200	0.025	0.025
0.200	0.025	0.025

NNM_cr	SDM_c	SDM_r
0.200	0.025	0.025
0.200	0.025	0.025
0.200	0.050	0.050
0.200	0.050	0.050
0.200	0.050	0.050
0.200	0.050	0.050
0.070	0.010	0.010
0.050	0.015	0.015
0.020	0.010	0.010
0.050	0.010	0.010
0.050	0.006	0.006
0.020	0.007	0.007
0.050	0.005	0.005
0.030	0.006	0.006
0.050	0.015	0.015
0.050	0.015	0.015
0.060	0.015	0.015
0.060	0.015	0.015
0.060	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.200	0.020	0.020
0.200	0.020	0.020
0.200	0.020	0.020

NNM_cr	SDM_c	SDM_r
0.200	0.025	0.025
0.200	0.025	0.025
0.200	0.025	0.025
0.200	0.025	0.025
0.050	0.015	0.015
0.050	0.015	0.015
0.050	0.015	0.015
0.060	0.015	0.015
0.060	0.015	0.015
0.060	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.050	0.015	0.015
0.050	0.015	0.015
0.050	0.015	0.015
0.100	0.015	0.015
0.100	0.015	0.015
0.200	0.025	0.025
0.200	0.025	0.025
0.200	0.025	0.025
0.200	0.025	0.025
0.300	0.025	0.025
0.300	0.025	0.025
0.300	0.025	0.025

NNM_cr	SDM_c	SDM_r
0.300	0.025	0.025
0.300	0.025	0.025
0.050	0.015	0.015
0.050	0.015	0.015
0.050	0.015	0.015
0.050	0.015	0.015
0.050	0.015	0.015
0.400	0.020	0.020
0.400	0.020	0.020
0.400	0.020	0.020
0.400	0.020	0.020
0.150	0.025	0.025
0.150	0.025	0.025
0.150	0.025	0.025
0.150	0.025	0.025
0.100	0.020	0.020
0.100	0.020	0.020
0.100	0.020	0.020
0.150	0.025	0.025
0.150	0.025	0.025
0.150	0.025	0.025
0.150	0.020	0.020
0.150	0.020	0.020
0.150	0.025	0.025
0.150	0.025	0.025

NNM_cr	SDM_c	SDM_r
0.100	0.020	0.020
0.100	0.020	0.020
0.150	0.017	0.017
0.150	0.017	0.017
0.005	0.006	0.006
0.005	0.006	0.006
0.020	0.006	0.006
0.020	0.006	0.006
0.050	0.009	0.009
0.050	0.009	0.009
0.100	0.010	0.010
0.100	0.010	0.010
0.120	0.010	0.010
0.120	0.010	0.010
0.200	0.015	0.015
0.200	0.015	0.015
0.300	0.025	0.025
0.300	0.025	0.025
2.500	0.200	0.200
2.500	0.200	0.200
0.015	0.015	0.015
0.030	0.015	0.015
0.015	0.015	0.015
0.020	0.010	0.010
0.010	0.015	0.015
0.010	0.015	0.015
0.100	0.050	0.050
0.020	0.020	0.020
0.100	0.050	0.050
0.020	0.010	0.010

NNM_cr	SDM_c	SDM_r
0.100	0.025	0.025
0.020	0.020	0.020
0.040	0.010	0.010
0.040	0.010	0.010
0.040	0.010	0.010
0.040	0.010	0.010
0.040	0.010	0.010
0.060	0.010	0.010
0.060	0.010	0.010
0.060	0.020	0.020
0.060	0.020	0.020
0.050	0.010	0.010
0.050	0.010	0.010
0.050	0.010	0.010
0.050	0.010	0.010
0.050	0.010	0.010
0.050	0.010	0.010
0.050	0.010	0.010
0.060	0.020	0.020
0.060	0.020	0.020
2.000	9.999	9.999
5.000	9.999	9.999

Table 12: OPF filter threshold parameters for the VIS channel

2.4 Transient flag handling for Irradiance Data

The GDPS also calculates the average of the SUN irradiance measurement that is acquired daily using instrument configuration ICIDs 8, 50 and 62. It is of paramount importance that pixels that are flagged as transient are **excluded** from the average. It has been demonstrated that single transient events can lead to 5% errors in the calculated averages.

3 Test results and examples

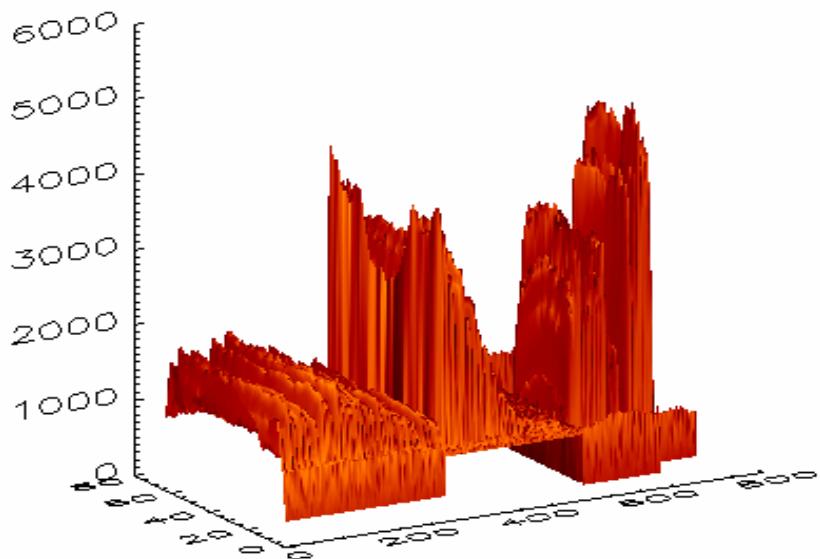
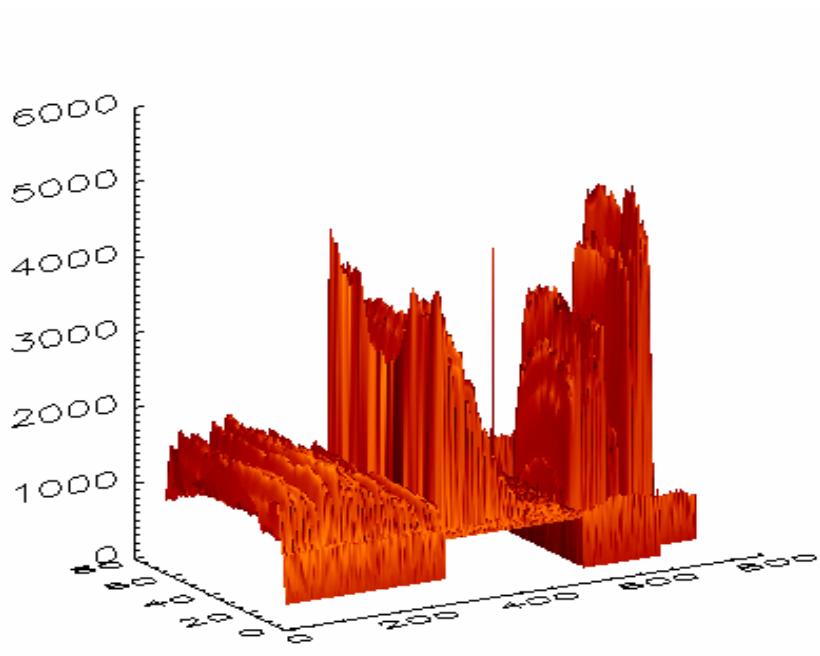


Figure 1: Top panel: solar irradiance measurement with spike. Bottom panel: same image but now with spike removed by the sequential difference method.

Appendix A IDL prototype source code

```

;-----  

; non radiance spike detection algorithm  

; version 1.0  

; Q.L. Kleipool  

; KNMI, 10-mei-2005 15:48  

;-----  
  

pro NonRadianceTransientDetection , Data , MeasurementClass , Settings  
  

;  

; determine the data dimensions  

;  
  

ColumnCount = n_elements(Data[*,0,0,0])  

RowCount = n_elements(Data[0,*,0,0])  

ChannelCount = n_elements(Data[0,0,*,0])  

FrameCount = n_elements(Data[0,0,0,*])  
  

;  

; declare some variables  

;  
  

SpikeMask = intarr(ColumnCount,RowCount,ChannelCount,FrameCount)  

Normalised = fltarr(ColumnCount,RowCount,ChannelCount,FrameCount)  

RowValues = fltarr(ColumnCount,RowCount,ChannelCount,FrameCount)  

ColValues = fltarr(ColumnCount,RowCount,ChannelCount,FrameCount)  
  

;  

; start dividing each frame with the previous frame  

; set the normalised values for the first frame to 1  

;  
  

for Channel=0,ChannelCount-1 do begin  

    for Frame=1,FrameCount-1 do begin  

        Normalised[*,*,Channel,Frame] = Data[*,*,Channel,Frame] /  

        Data[*,*,Channel,Frame-1]  

        endfor  

        Normalised[*,*,Channel,0] = 1.  

    endfor

```

```

;

; check the measurement class and set the subchannel handling accordingly
; channel 0 is the uv channel that has two sub-channels UV2 and UV1. (note the
; correct sequence!)
; channel 1 is the vis channel which has no sub-channels
; these sub-channels are only used for the following measurement classes
;

case MeasurementClass of
  0 : MaxSubChannels=2          ; earth
  1 : MaxSubChannels=2          ; sun
  2 : MaxSubChannels=2          ; wls
  3 : MaxSubChannels=1          ; led
  4 : MaxSubChannels=1          ; dark
else : print,'unknown measurement class: ',MeasurementClass
endcase

;

; detect the spikes
; using nearest neighbour filtering for the first frame
; and the sequential difference algorithm for all following frames
; store the calculated spike-levels for tuning purposes
; note that for the first frame the spike levels are equal in the row and column
; direction
;

for Channel=0,ChannelCount-1 do begin
  SubChannelCount = long(Channel EQ 0 ? MaxSubChannels : 1)
  SplitColumn      = long(Channel EQ 0 AND MaxSubChannels EQ 2 ? 615 : ColumnCount)
  for SubChannel=0,SubChannelCount-1 do begin
    StartColumn = long(SubChannel EQ 0 ? 0 : SplitColumn)
    StopColumn  = long(SubChannel EQ 0 ? SplitColumn : ColumnCount)
    StopColumn  = StopColumn - 1
    if Channel EQ 0 AND SubChannel EQ 0 then
      SubChannelSettings=Settings[where(settings.Channel EQ 'uv2')]
    if Channel EQ 0 AND SubChannel EQ 1 then
      SubChannelSettings=Settings[where(settings.Channel EQ 'uv1')]
    if Channel EQ 1 AND SubChannel EQ 0 then
      SubChannelSettings=Settings[where(settings.Channel EQ 'vis')]
    print,SubChannelSettings,StartColumn,StopColumn,SplitColumn
    Frame = 0
    Slice = reform(Data[StartColumn:StopColumn,*,Channel,Frame])
  
```

```
SpikeMask[StartColumn:StopColumn,* ,Channel,Frame] =  
NearestNeighbour(Slice,SubChannelSettings,COST=Cost)  
  
RowValues[StartColumn:StopColumn,* ,Channel,Frame] = Cost  
ColValues[StartColumn:StopColumn,* ,Channel,Frame] = Cost  
for Frame=1,FrameCount-1 do begin  
    Slice = reform(Normalised[StartColumn:StopColumn,* ,Channel,Frame])  
    SpikeMask[StartColumn:StopColumn,* ,Channel,Frame] =  
DetectTransient(Slice,SubChannelSettings,ROWVALUES=RowVal,COLVALUES=ColVal)  
    ColValues[StartColumn:StopColumn,* ,Channel,Frame] = ColVal  
    RowValues[StartColumn:StopColumn,* ,Channel,Frame] = RowVal  
endfor  
endfor  
endfor  
  
save,Data,MeasurementClass,SpikeMask,ColValues,RowValues,Settings,FILE='result.sav'  
end  
  
;-----
```

```

;-----  

;  

; nearest neighbour comparison algorithm  

;  

; used by the transient algorithm to detect spikes in the first frame of a  

; time-series of frames  

; not as accurate as the sequential difference algorithm but should be sufficient  

; when enough data is available  

; compares each pixel with the median value of all the surrounding pixels,  

; including itself  

; for a median filter it does not matter when the central pixel is included  

; the width of the neighbourhood can be controlled in both direction through the  

; settings variable  

; setting these widths both to zero results in no flagging at all  

; setting these widths both equal to one result in the traditional nearest  

; neighbour method  

; returns a frame with the same size as the input data containing flags for each  

; pixel that contains a spike  

;  

; KNMI, 10-mei-2005 16:03  

; version 1.1  

; Q.L. Kleipool  

;  

;-----  

function NearestNeighbour , Data , Settings , COST=Cost  

ColumnCount = n_elements(Data[*,0])  

RowCount = n_elements(Data[0,*])  

Threshold = Settings.nn_threshold  

ColumnSmoothLength = Settings.nn_width_c < ColumnCount  

RowSmoothLength = Settings.nn_width_r < RowCount  

SpikeMask = intarr(ColumnCount,RowCount)  

Cost = fltarr(ColumnCount,RowCount)  

SpikeFlag = 1  

for Row=0,RowCount-1 do begin  

  for Column=0,ColumnCount-1 do begin  

    StartColumn = long((Column - ColumnSmoothLength) > 0)
  
```

```
StopColumn = long((Column + ColumnSmoothLength) < (ColumnCount-1))
StartRow   = long((Row - RowSmoothLength) > 0)
StopRow    = long((Row + RowSmoothLength) < (RowCount-1))
Value      = median(Data[StartColumn:StopColumn,StartRow:StopRow])
SpikeLevel = (Data[Column,Row] / Value) - 1
Flag       = SpikeLevel GT Threshold
SpikeMask[Column,Row] = Flag * SpikeFlag
Cost [Column,Row]      = SpikeLevel
endfor
endfor

return,SpikeMask
end

; -----
```

```
;-----  
; sequential difference spike detection algorithm  
; version 1.0  
; Q.L. Kleipool  
; KNMI, 10-mei-2005 15:48  
;-----  
  
function DetectTransient , Data , Settings , ROWVALUES=RowValues ,  
COLVALUES=ColValues  
  
;  
; check the data size  
;  
  
ColumnCount = n_elements(Data[*,0])  
RowCount = n_elements(Data[0,*])  
  
;  
; declare some variables  
;  
  
SpikeMaskRow = intarr(ColumnCount,RowCount)  
SpikeMaskCol = intarr(ColumnCount,RowCount)  
RowValues = fltarr(ColumnCount,RowCount)  
ColValues = fltarr(ColumnCount,RowCount)  
Cost = fltarr(ColumnCount,RowCount)  
  
;  
; extract the requested parameters  
;  
  
RowSmoothLength = Settings.width_r  
ColSmoothLength = Settings.width_c  
Threshold = Settings.threshold  
  
;  
; start applying additional smoothing in the column direction  
; if the column smooth width is zero or one do nothing  
; if the column smooth width is less then zero use a median filter over the full  
; range  
; if the column smooth width is greater then one use a median filter of the  
; specified width
```

```

;

for Row=0,RowCount-1 do begin
  ThisRow = reform(Data[*,Row])
  if RowSmoothLength LT 0 then RowFit = median(ThisRow)
  if RowSmoothLength EQ 0 then RowFit = fltarr(ColumnCount) + 1.
  if RowSmoothLength EQ 1 then RowFit = fltarr(ColumnCount) + 1.
  if RowSmoothLength GT 1 then RowFit = median(ThisRow,RowSmoothLength)
  RowValues[*,Row] = (ThisRow / RowFit) - 1
endfor

;

; start applying additional smoothing in the row direction
; if the row smooth width is zero or one do nothing
; if the row smooth width is less then zero use a median filter over the full range
; if the row smooth width is greater then one use a median filter of the specified
; width
;

for Col=0,ColumnCount-1 do begin
  ThisCol = reform(Data[Col,*])
  if ColSmoothLength LT 0 then ColFit = median(ThisCol)
  if ColSmoothLength EQ 0 then ColFit = fltarr(RowCount) + 1.
  if ColSmoothLength EQ 1 then ColFit = fltarr(RowCount) + 1.
  if ColSmoothLength GT 1 then ColFit = median(ThisCol,ColSmoothLength)
  ColValues[Col,*] = (ThisCol / ColFit) - 1
endfor

;

; determine where the thresholds have been violated
;

if RowSmoothLength NE 0 AND RowSmoothLength NE 1 then begin
  RowMask = RowValues GT Threshold
endif else begin
  RowMask = intarr(ColumnCount,RowCount) + 1
endelse

if ColSmoothLength NE 0 AND ColSmoothLength NE 1 then begin
  ColMask = ColValues GT Threshold
endif else begin

```

```
ColMask = intarr(ColumnCount,RowCount) + 1
endelse

;

; now combine the mask to yield the final spike mask
; a spike is flagged if it has been detected in both directions
;

Result = ColMask AND RowMask

return,Result
end

;-----
```